Ministry of Education and Science of Ukraine

National Technical University of Ukraine

«Kyiv Polytechnic Institute. Igor Sikorsky »

Faculty of Informatics and Computer Technologies

Department of Computer Engineering

LAB № 8

from the discipline "Theory of Algorithms"

on the topic «Dynamic programming»

PERFORMED BY:

1st year student

group ІП-93

Dominskyi V.O.

The credit - 9311

Variant – 11

CHECKED:

Associate Professor of OT

c.t.s.,s.r.

Antoniuk А.І.

Kyiv - 2020

**TASK**

**Goal:** implementation of the dynamic programming algorithm for the backpack problem.

**Option task:**Given are n different items known about their size, or weight, *wi*, and cost *vi*. There is a backpack in which to put these items. The backpack is known for its capacity (total size, or weight, of items that can be placed in the backpack) - W. It is necessary to select a set of items S among all given items that (1) their total dimension does not exceed the capacity of the backpack W and (2) the total value of the objects in the set S is the highest possible among all other sets.

**CODE**

**using** System**;**

**public** class BackPackClass

**{**

//main function

**public** static void Main**()**

**{**

int **[]value** **=** **new** int**[]{**20**,** 220**,** 300**};**

int **[]**weight **=** **new** int**[]{**20**,** 40**,** 30**};**

int W **=** 50**;**

int n **=** 3**;**

Console**.**WriteLine**(**BackPack**(**W**,** weight**,** **value,** n**));**

**}**

//Function that returns

//maximum of two numbers

static int max**(**int a**,** int b**)**

**{**

**if(**a**>**b**)**

**{**

**return** a**;**

**}**

else

{

return b;

}

}

// Returns the maximum value that can

// be put in a knapsack of capacity W

static int BackPack(int W, int []weight, int []value, int n)

{

//if something went really wrong

if (n == 0 || W == 0)

{

return 0;

}

// If weight of the n item is

// more than Backpack W,

// then this item is bad

if (weight[n-1] > W)

{

return BackPack(W, weight, value, n-1);

}

// Return the maximum of two cases:

//1. n item included

//2. not included

else

{

return max( value[n-1] +

BackPack(W-weight[n-1], weight, value, n-1),

BackPack(W, weight, value, n-1));

}

}

}

**RESULTS OF THE PROGRAM WORK**

The input arrays are:

value = new int[]{20, 220, 300};

weight = new int[]{20, 40, 30};

W = 50

Output array:

320

**CONCLUSIONS**

I got acquainted with the topic of laboratory work.

Have acquired relevant work skills.

An appropriate test program has been developed.

Learn the history of creating this task. Considered several solutions, namely through dynamic programming.